

GSAW 2005

March 28, 2005

An Introduction to Message-Bus Architectures for Space Systems

½ Day Tutorial Proposal

Dan Smith
NASA Goddard Space Flight Center
MS 581.0
Greenbelt, Maryland 20771

301-286-2230
dan.smith@nasa.gov

Brian Gregory
Interface & Control Systems, Inc.
8945 Guilford Rd
Columbia, Maryland 21046

1-877-808-2668
briang@interfacecontrol.com

ABSTRACT

This course presents technical and programmatic information on the development of message-based architectures for space mission ground and flight software systems. Message-based architecture approaches provide many significant advantages over the more traditional socket-based one-of-a-kind integrated system development approaches. The course provides an overview of publish/subscribe concepts, the use of common isolation layer API's, approaches to message standardization, and other technical topics. Several examples of currently operational systems are discussed and possible changes to the system development process are presented. Benefits and lessons learned will be discussed and time for questions and answers will be provided.

An Introduction to Message-Bus Architectures for Space Systems

A ½ Day GSAW 2005 Tutorial

Tutorial Description:

This course presents technical and programmatic information on the development of message-based architectures for space mission ground and flight software. Message-based architecture approaches provide many significant advantages over the more traditional socket-based one-of-a-kind integrated system development approaches.

Two instructors will co-teach the half-day class. Dan Smith has been developing satellite ground systems for 25 years and currently manages the message-bus architecture efforts at NASA's Goddard Space Flight Center (GSFC). Brian Gregory has over ten years of experience with middleware-based control architectures. He works for Interface & Control Systems and has supported numerous missions at NASA/GSFC. He is currently one of the lead API developers on the GSFC Program managed by Dan Smith.

This course presents technical and programmatic information on the development of message-based architectures for space mission ground and flight software systems. Message-based architecture approaches provide many significant advantages over the more traditional socket-based one-of-a-kind integrated system development approaches. Message-base architectures allows products/components from different vendors, or from in-house developments, to "plug and play" together, greatly reducing system integration time and allowing for greater choice in component selection. Technology infusion over time is simplified when new components can be added without impacting components already on the bus. The course provides an overview of publish/subscribe concepts, the use of common isolation layer API's, approaches to message standardization, and other technical topics. Several examples of currently operational systems are discussed and possible changes to the system development process are presented. Benefits and lessons learned will be discussed and time for questions and answers will be provided.

Course Outline:

1. Introduction

2. Technical Topics

- Publish/Subscribe Technical Description
- Application Programming Interface (API) isolation layer
- Development and use of standardized messages
- Characteristics of a Good Software Component
- Performance, Reliability
- System Integration

3. Operational Examples

- EO-1 using the ICS Message Bus
- Recent NASA/GSFC Mission Control Reengineering Efforts

4. Affects on the Development Process

- Ability to use multiple small development teams
- Reduced team interface interaction
- "Shopping for Components"
- Changes to the System Test Activity

5. Summary of Benefits and Lessons Learned

6. Open Questions

What participants should expect to learn:

Participants should gain a basic understanding of the technical concepts involved with the creation and development of message-based architectures for spacecraft ground and flight systems. Steps to create a reference architecture will be shown and lessons learned from NASA and industry efforts will be discussed. Participants should also gain an appreciation for the potential significant long-term benefits possible with these types of architectures.

Intended participants and prerequisite knowledge:

Participants should have a basic understanding of ground systems. In addition, it is helpful to have some software development background or software development process understanding.

Expected/desired number of participants: 8 - 25

List of materials to be provided: Copy of course materials.

Biography (100-200 words):

Dan Smith has 25 years of experience developing satellite ground systems. He was the technical lead on the early Hubble Space Telescope mission control center, the Program Manager for NOAA's 5-satellite GOES weather satellite control system, and chief architect for Globalstar's constellation control center which now handles 52 satellites and nearly 2000 satellite contacts per day. Mr. Smith became a NASA employee in September 2001 with an assignment to infuse commercial practices and satellite constellation concepts into NASA. His "GMSEC" architecture uses standardized messaging to allow any of a large set of functional components to be easily integrated in a "configure-and-go" manner to support a wide variety of current and planned NASA missions. He has an MS degree from George Washington University and has taught software courses at the University of Maryland and George Washington University.

Brian Gregory has been engineering satellite systems for more than 10 years with Interface & Control Systems, Inc. Mr. Gregory has been a lead member of the product development team for Interface & Control System's Commercial Off the Shelf (COTS) SCL (Spacecraft Command Language) software system. SCL uses the message bus architecture to provide a distributed and scalable system for both flight and ground automation. Mr. Gregory's has been integral in the development of the ICS "Software Bus", a messaging abstraction layer that allows the integration of various Message Oriented Middlewares (MOMs) with the SCL COTS solution without core software refactoring. In 1998, he participated in the development of the FUSE (Far Ultraviolet Spectroscopic Explorer) ground system that used the SCL messaging architecture to simplify the transition from integration & test to flight operations. In 2002, he participated in the reengineering of NASA's EO-1 that used messaging to integrate the legacy flight software with SCL's expert system. Recently he has been applying his experience to the development of the GMSEC messaging Application Programming Interface (API), a generic messaging layer designed to facilitate the fast and easy integration of components using GMSEC standards.

Has this tutorial been given before: NO